

第 17 课：多态之抽象类语法

讲师：Monkey

时间：2017 年 5 月 7 日（周日）

课程要点：

1.抽象方法

2.抽象类

3.使用场景

1.抽象方法

1.虚方法->抽象方法

父类里面用 `virtual` 关键字修饰的方法叫做虚方法，子类可以使用 `override` 重新该虚方法，也可以不重写。

虚方法还是有方法体的，当我们父类中的这个方法已经虚到完全无法确定方法体的时候，就可以使用另外一种形式来表现，这种形式叫抽象方法。

2.抽象方法语法

抽象方法的返回值类型前用关键字 `abstract` 修饰，且无方法体。

抽象方法必须存在于抽象类中。

2.抽象类

1.抽象类语法

在定义类的关键字 `class` 前面加 `abstract` 修饰的类就是抽象类。

子类继承抽象类，使用 `override` 关键字重写父类中所有的抽象方法。

2.抽象类注意事项

<1>抽象类中不一定要有抽象方法，但是抽象方法必须存在于抽象类中。

<2>抽象类不能被实例化，因为抽象类中有抽象方法(无方法体)，如果真能实例化抽象类的话，调用这些无方法体的方法是没有任何意义的，所以无法实例化。

3.使用场景

<1>当父类中的方法不知道如何去实现的时候，可以考虑将父类写成抽象类，将方法写成抽象方法。

<2>如果父类中的方法有默认实现，并且父类需要被实例化，这时可以考虑将父类定义成一个普通类，用虚方法实现多态。

<3>如果父类中的方法没有默认实现，父类也不需要被实例化，则可以将该类定义为抽象类。